Lecture 17

Space lower bounds for data stream algorithms

Course: Algorithms for Big Data

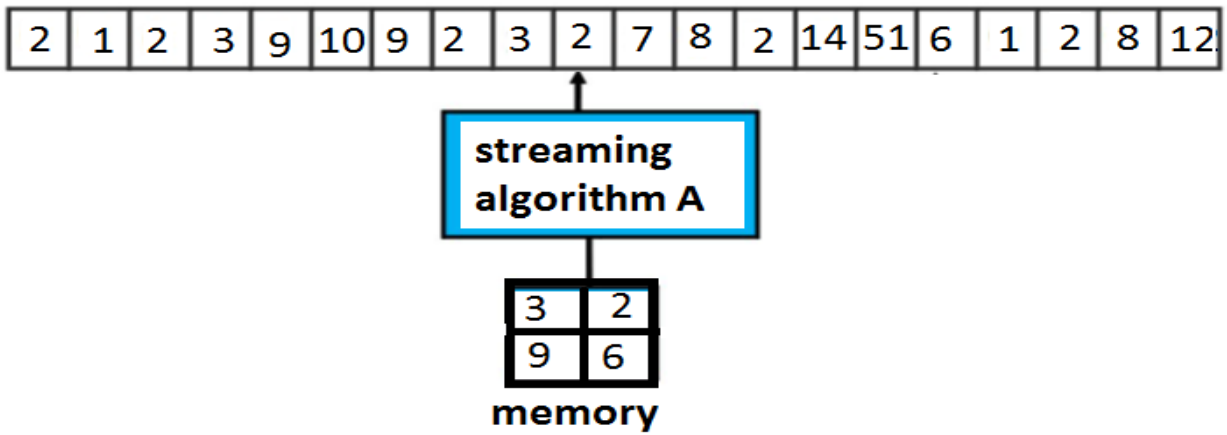
Instructor: Hossein Jowhari

Department of Computer Science and Statistics
Faculty of Mathematics
K. N. Toosi University of Technology

Spring 2021

framework for getting space lower bounds

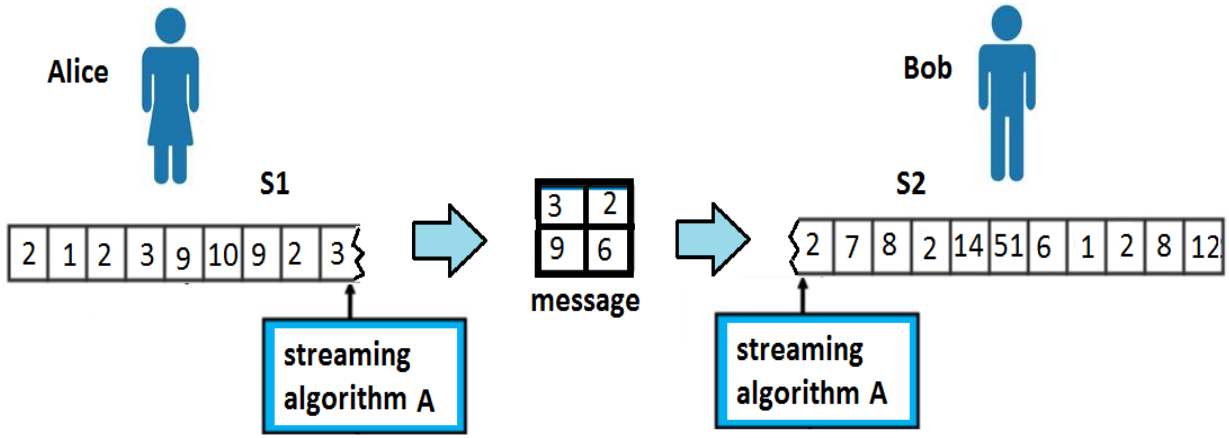
Suppose we have a streaming algorithm A that computes the function $f(S)$ where S is a finite stream $S = s_1, \ldots, s_m$



We assume the function f is defined for streams of every length.

framework for getting space lower bounds

Suppose we split the stream into two parts. We give the first part S_1 to Alice and give Bob the second part S_2.



Alice runs algorithms A on her part and then sends the content of the memory to Bob. Bob knows the algorithm A. He resumes the computation on his part.

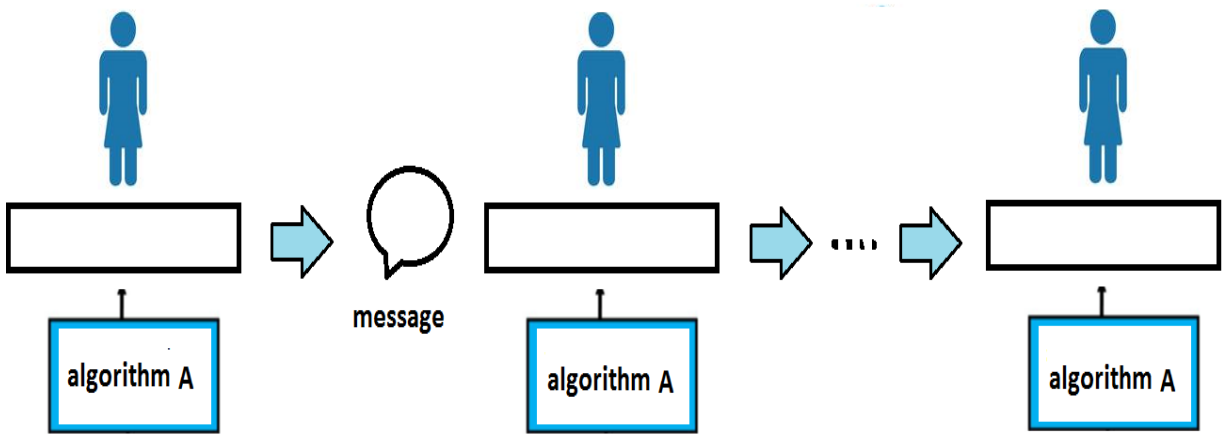
In the end, Bob has computed the value of function f on the stream $S = S_1 S_2$.

framework for getting space lower bounds

Alice and Bob now have a Communication Protocol for computing the value of $f(S)$ where the first part of S is given to Alice and the second part is given to Bob.

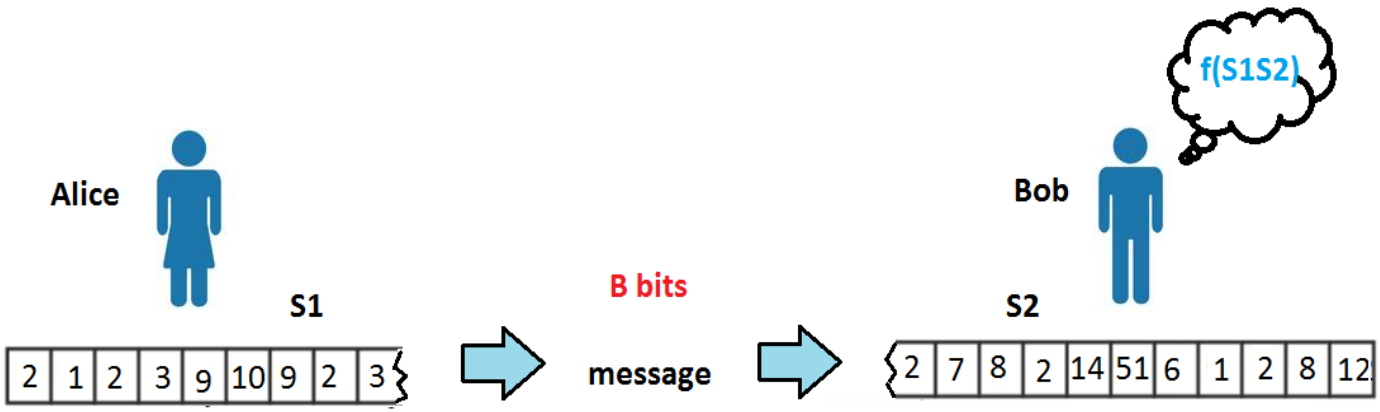
Communication protocols can be deterministic or randomized.

It is a communication with 2 players. We could have communication protocols with any number of players. A streaming algorithms is like a (one-way) communication protocol with m players. Here m is the length of the stream.



framework for getting space lower bounds

Consider the 2-player one-way setting. Suppose the players compute the value of function $f(S)$ where $S = S_1 S_2$.



Suppose we know every one-way communication protocol for computing the value of $f(S)$ requires at least B bits of communication. This means, in the most efficient protocol for f, there is an input where the size of the transmitted message from Alice to Bob is at least B bits.

We conclude that every streaming algorithm for computing $f(S)$ requires B bits of memory!

framework for getting space lower bounds

Main Conclusion: Therefore a lower bound for the size of the message in the 2-player 1-way communication protocols that computes $f(S)$ will be a lower bound for the space complexity of the streaming algorithms that compute $f(S)$.

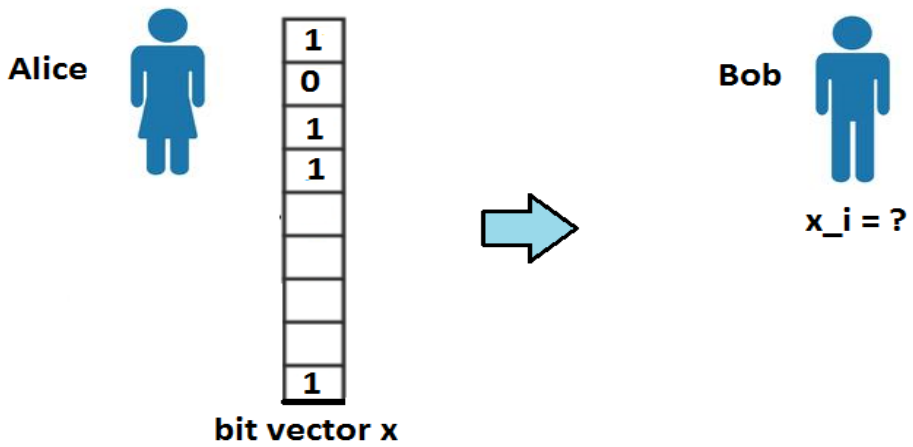
As result, in order to find a lower bound for the space complexity of streaming algorithm for $f(S)$ we can consider the problem in the two player setting.

Since we are interested in lower bounds (impossibility results), often it would be enough to find a special case of the problem that is hard enough.

Also we use reductions between problems to prove a lower bound for our desired function f.

An example: Indexing Problem

In an instance of Indexing Alice has a vector $\boldsymbol{x} \in \{0,1\}^n$ and Bob has an index $i \in \{1,\ldots,n\}$. Bob wants to know the value of x_i.



Theorem Every randomized communication protocol for Indexing over n bits that succeeds with probability $3/4$ requires a message size of $\Omega(n)$ bits.

We say the randomized communication complexity of indexing is $\Omega(n)$.

A lower bound for estimating F_∞

Recall that for a frequency stream, F_∞ is the number of the repetitions of the most frequent element in the stream.

$$F_\infty = \max_{i=1}^{n} f_i$$

Here f_i is the frequency of the element i.

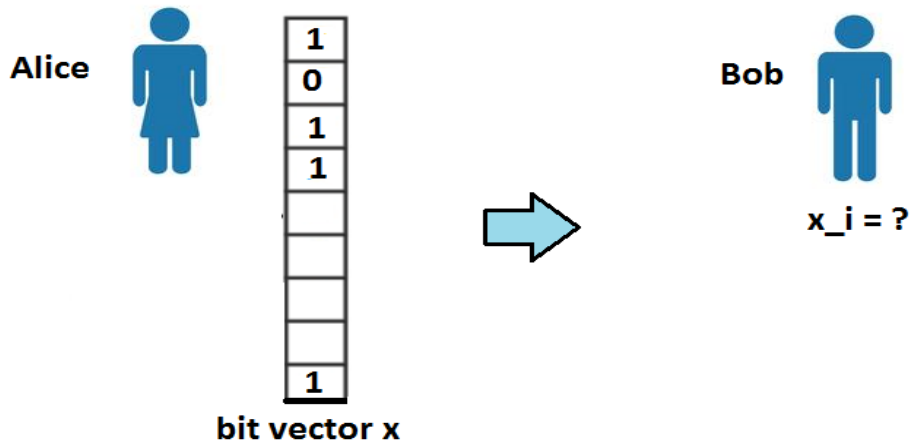
Example: Stream $= 1, 2, 2, 2, 1, 4, 5, 6, 8, 9, 10, 13, 2, 2, 1, 4$

$$F_\infty = 5$$

Lemma: Any randomized streaming algorithm for approximating F_∞ within a factor better than $1/2$ requires $\Omega(n)$ space.

We use the communication lower for Indexing to prove a space lower bound for estimating F_∞.

Consider the Indexing problem.



Alice transforms her bit vector into a sequence of numbers. For all $r \in [n]$, if $x_r = 1$ Alice adds r to her sequence. For example if the bit-vector is $\boldsymbol{x} = (0, 1, 1, 0, 1, 1, 0, 1, 0, 1)$ Alice's sequence would be $S = 2, 3, 5, 6, 8, 10$

On the other hand, Bob has number i (his input). Suppose $i = 8$.

Now suppose we append the number i to the end of Alice's sequence. We get the new sequence $T = S, i$. In our example the new sequence would be

$$T = 2, 3, 5, 6, 8, 10, \quad 8$$

It is easy to see that if $x_i = 1$ then $F_\infty(T)$ would be 2 otherwise it will be 1.

A streaming algorithm for estimating F_∞ with approximation factor better than $1/2$ can distinguish between these two cases. As result we can use such a streaming algorithm to solve the Indexing problem.

Therefor we established the statement of our lemma.

Problem: Find a space lower bound of $\Omega(n)$ for testing connectivity of graph $G = (V, E)$ when the input stream is an arbitrary ordering the edges E.

Is graph G connected or not?